Adaptive Multiscale Predictions at Exascale

Assad Oberai, Onkar Sahni & Mark Shephard
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180

Motivation Predictive simulation of problems that span multiple scales is a grand challenge in applied mathematics and computational science. For these problems the simplistic approach of resolving all spatio-temporal scales has not been feasible on the present-day petascale platforms. Nor will it be facilitated by the proposed advances to exascale computing. A point in case is the problem of simulating phase transitions and thermal transport in a microchannel. This problem involves phenomena like cavitation, moving contact lines and rupture of thin films, that occur at molecular scales ($10^{-10}m$ and $10^{-14}s$), and others, such as the attainment of critical heat flux, that occur at the scale of the device ($10^{-2}m$ and $10^{-2}s$). A simplistic treatment of this problem in three dimensions will lead to 10^{24} grid points and 10^{12} time steps. This is the cost for a single simulation, and it is clear that even this is far beyond the computing power of the proposed exascale platforms. However, reliable predictions require a multitude of such simulations in order to account for the associated uncertainties. There are several such multiscale problems in science and engineering like turbulence, climate modeling, etc., where a simplistic approach will not succeed. What is required is a strategy that is based on the judicious choice of modeling at the scale that is necessary to attain a certain predictive goal. **That is, multiscale algorithms that are based on some form of adaptive error control and uncertainty management. This white paper is concerned with the advances in applied mathematics that will be necessary to develop these algorithms for the proposed exascale platforms.**

Adaptive Error Control We interpret adaptive error control and uncertainty management in a broad context. It is applied to the:

1. Control of discretization in space and time domain within a single type of modeling approach (continuum, mesoscale, atomistic, quantum etc.).
2. Control of multiscale coupling strategy. Whether one selects a simple sequential approach or a more accurate, but expensive, concurrent approach.
3. Control of the extent of space-time coupling domains within a concurrent strategy.
4. Control of discretization in stochastic space.
5. Selection of models driven by evidence and plausibility.

Exascale Challenges While the precise architecture of a typical exascale platform is still a matter of research, it is becoming clear that it will involve hybrid computing strategies, deep memory hierarchies, a premium on the power consumed for a calculation and movement of data, and algorithms that are fault tolerant. It is therefore important to determine what advances are required to achieve these targets in an adaptive multiscale simulation. A brief, and by no means exhaustive, list is presented below:

1. The execution of adaptive mutliscale computations on exascale computers must first address the fact that we need to define the mathematical infrastructure for defining scale linking techniques and discretization errors associated with them. We must then consider the algorithms for their implementation on exascale computers that **minimize the**

movement of data on machines with very limited memory per compute resource. On a broader level, the mathematical framework of mutliscale adaptation, and the algorithms that implement it, must account for the characteristics of exascale computers. For example, on current systems the effective use of accelerators requires the movement of the data into the specific memory infrastructure of the accelerator. This places a large penalty on adaptive methods that would execute adaptive control on a CPU and then move the data for computation to an accelerator. However, if advances in “moving the computation to the data” become a reality in some effective form, this penalty may be eliminated or drastically reduced. Clearly the forms of adaptive multiscale control that will be most effective will be different in the two cases and, in the second case, be a function of how the “computation is moved to the data”.

2. **Fast and efficient dynamic load balancing** is an essential part of the implementation of the adaptive multiscale algorithms. Although it is simple to state that the goal of the dynamic load balancing process is to minimize the power requirements, the goal is in fact a dynamic multi-objective, multi-parameter optimization problem. The objectives must account for the amount of memory used, amount of data movement through a heterogeneous hierarchy, and the execution of computations on heterogeneous processing units; all of which account for constraints such as completion of the entire simulation in a acceptable wall clock time. The dynamic load balancing algorithms must be able to map the set of parameters that characterize the parallel computer to methods that are charged with load balancing heterogeneous calculations (e.g., atomistics and mesh-based PDE). In addition, since the adaptive control processes and dynamic load-balancing will be executed on a regular basis, their “cost” must be accounted for when deciding when to adapt, how to adapt, and how to dynamically load balance.
3. In the context of multi-level algorithms one way of achieving **fault tolerance** involves separating the overall algorithm into inner and outer algorithms. The outer algorithm is designed so that it can accommodate faults in the inner algorithm, and it is assumed that the outer algorithm will be, relatively speaking, fault-free. This leads to an efficient strategy since most of the computational effort in these algorithms is expended in the inner iterations which can be performed using compute cycles that are not robust. A multiscale algorithm naturally provides this inner/outer framework. The challenge then is to re-cast typical multiscale algorithms into this framework while in the process ensuring that the structure of the outer loop is reliable with respect to faults at all levels (including its own).
4. Uncertainty quantification leads to matrices with dense block structure and sparse sub-blocks, or sparse block structure with dense sub-blocks. This choice opens up the possibility of mapping data storage to the hierarchic memory storage on exascale computers with the goal of **minimizing communication**.